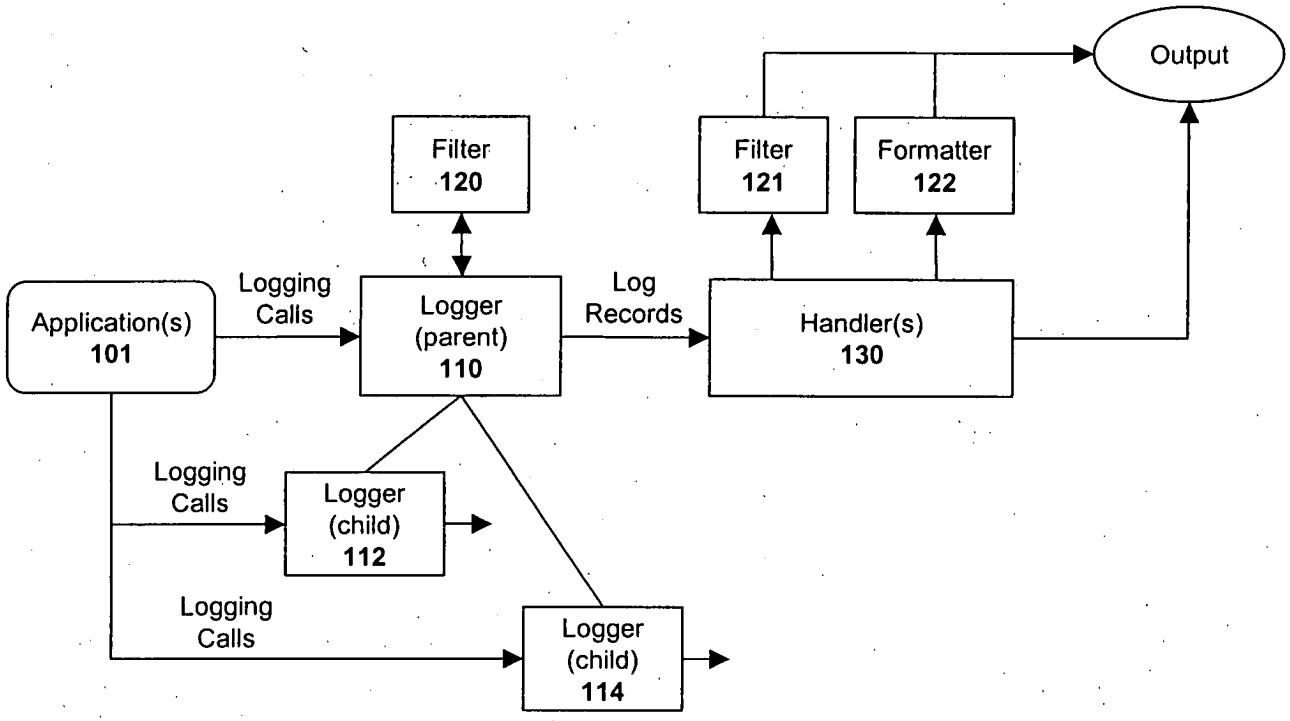

Fig. 1

(Prior Art)

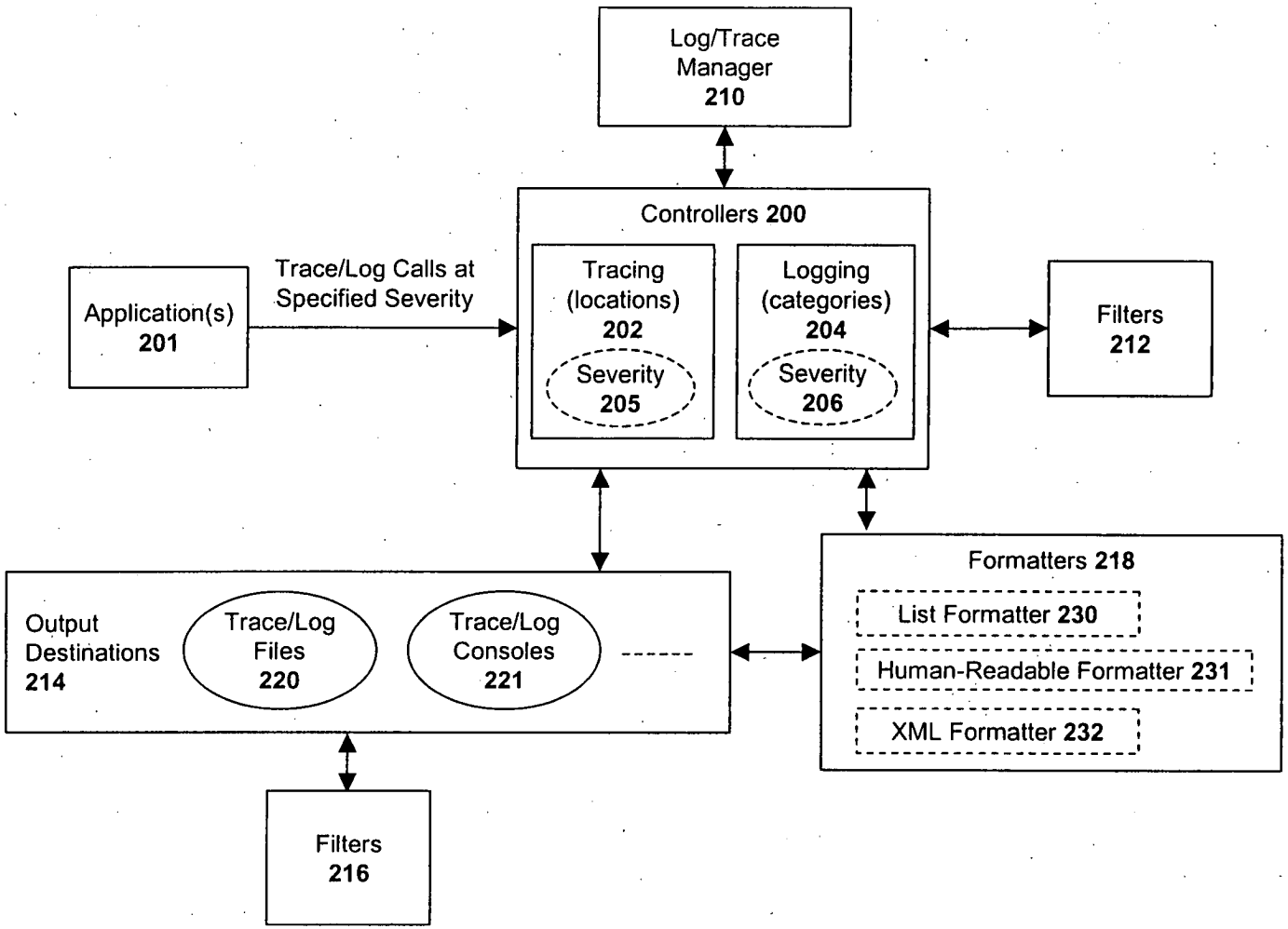
Log/Trace
Manager
210
Controllers 200
Tracing
(locations)
202
Severity
205
Logging
(categories)
204
Severity
206
Application(s)
201
Trace/Log Calls at
Specified Severity
Filters
212
Formatters 218
List Formatter 230
Human-Readable Formatter 231
XML Formatter 232
Output
Destinations
214
Trace/Log
Files
220
Trace/Log
Consoles
221
Filters
216

Fig. 2

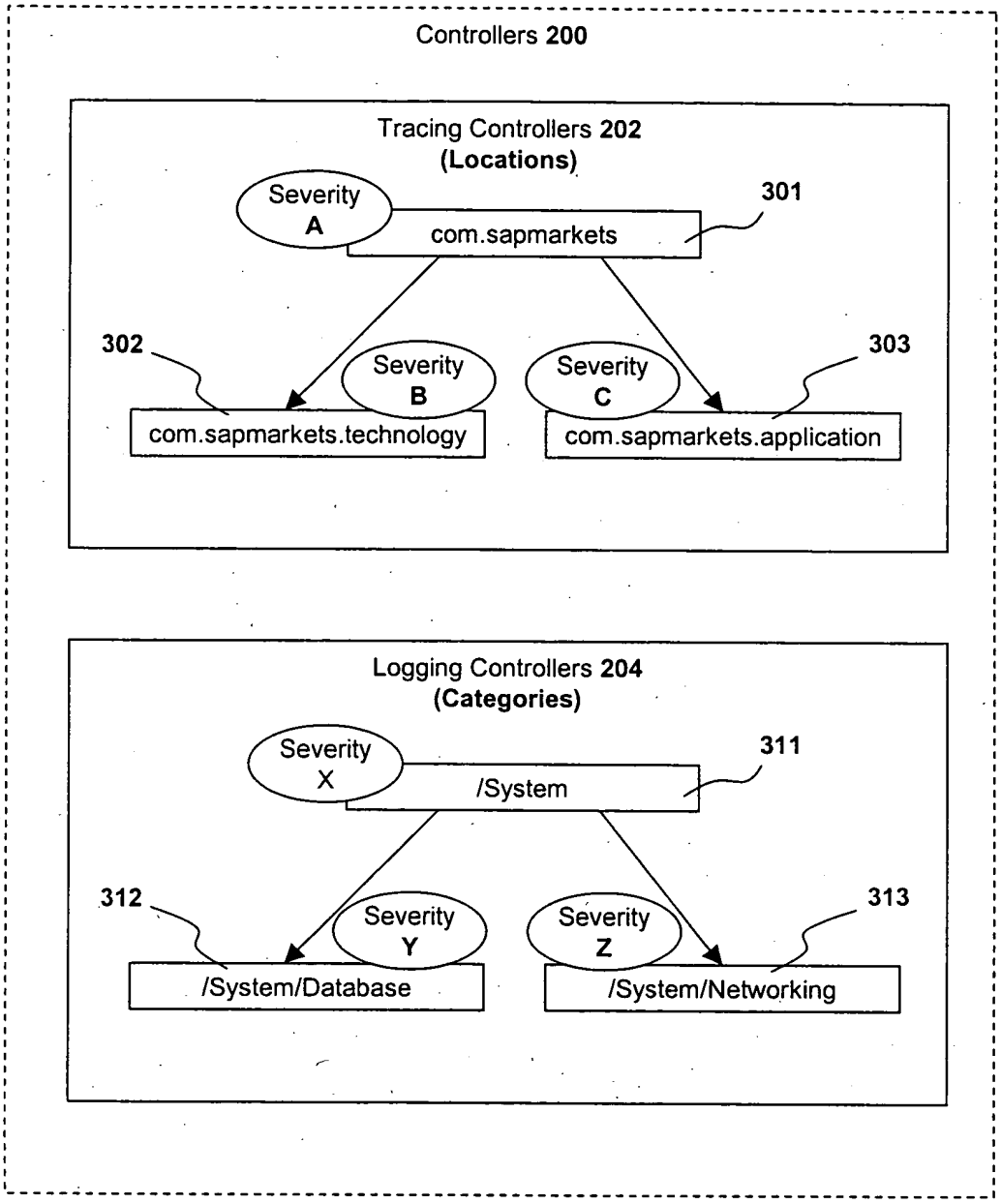
Controllers 200
Tracing Controllers 202
(Locations)
Severity
A
301
com.sapmarkets
302
Severity
B
303
Severity
C
com.sapmarkets.technology
com.sapmarkets.application
Logging Controllers 204
(Categories)
Severity
X
311
/System
312
Severity
Y
313
Severity
Z
/System/Database
/System/Networking

Fig. 3

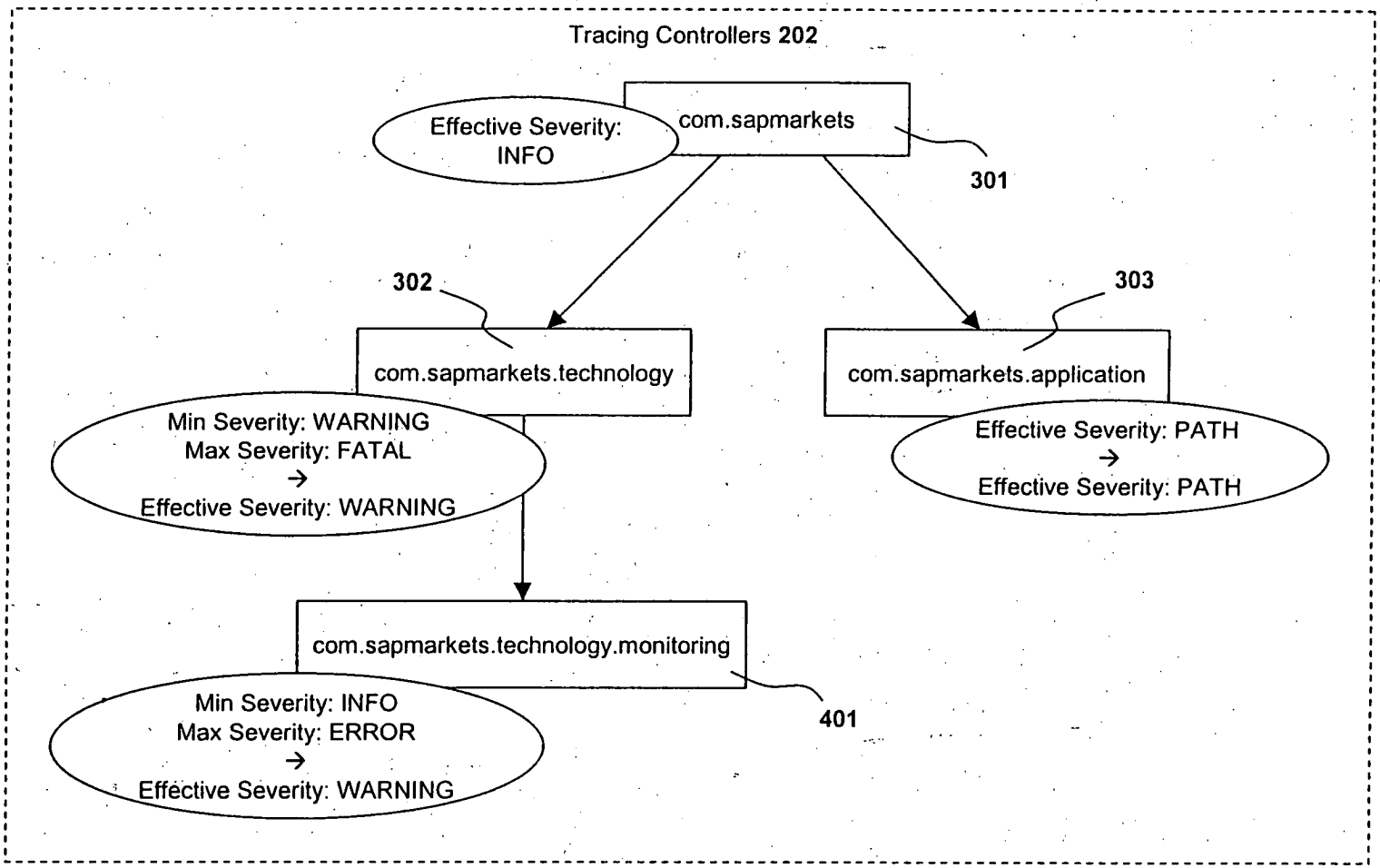
Tracing Controllers 202
Effective Severity: INFO
com.sapmarkets
301
302
303
com.sapmarkets.technology
com.sapmarkets.application
Min Severity: WARNING
Max Severity: FATAL
→
Effective Severity: WARNING
Effective Severity: PATH
→
Effective Severity: PATH
com.sapmarkets.technology.monitoring
401
Min Severity: INFO
Max Severity: ERROR
→
Effective Severity: WARNING

Fig. 4

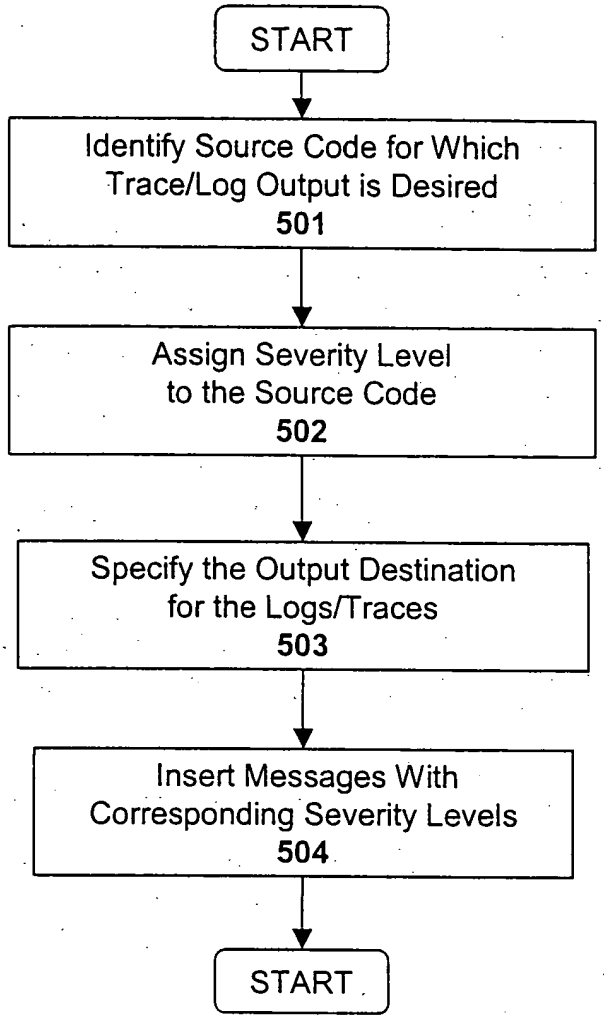
START
Identify Source Code for Which Trace/Log Output is Desired
501
Assign Severity Level to the Source Code
502
Specify the Output Destination for the Logs/Traces
503
Insert Messages With Corresponding Severity Levels
504
START

Fig. 5

MIN	0	Minimum restriction
DEBUG	100	For debugging purpose, with extensive and low level information
PATH	200	For tracing the execution flow, e.g., used in the context of entering and leaving a method, looping and branching operations
INFO	300	Informational text, mostly for echoing what has been performed
WARNING	400	Application can recover from anomaly, and fulfill the desired task, but need attention from developer/operator
ERROR	500	Application can recover from error, but cannot fulfill desired task due to the error
FATAL	600	Application cannot recover from error, and the severe situation causes fatal termination
MAX	700	Maximum restriction

Fig. 6a

ALL	0	Output messages of all severity
NONE	701	No messages will be logged

Fig. 6b

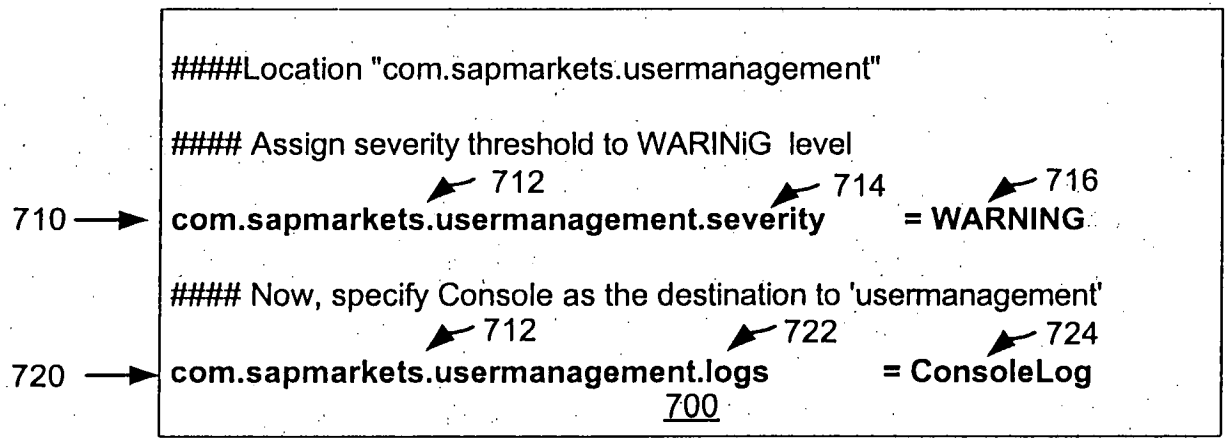
####Location "com.sapmarkets.usermanagement"
Assign severity threshold to WARINiG level
712
714
716
710
com.sapmarkets.usermanagement.severity = WARNING
Now, specify Console as the destination to 'usermanagement'
712
722
724
720
com.sapmarkets.usermanagement.logs = ConsoleLog
700

FIG. 7a

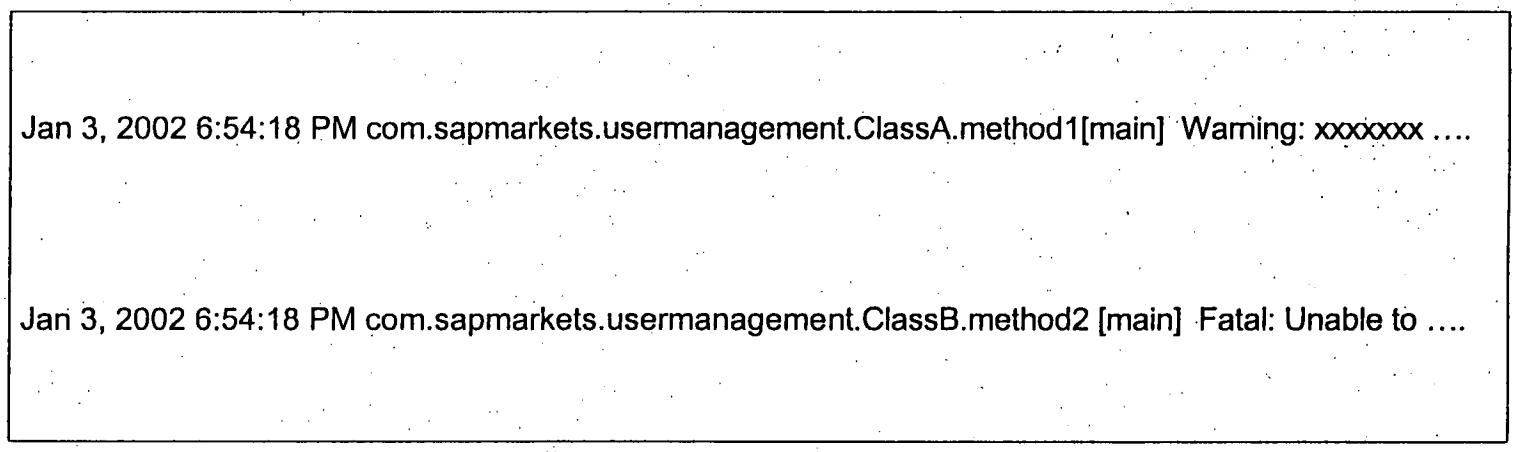
Jan 3, 2002 6:54:18 PM com.sapmarkets.usermanagement.ClassA.method1[main] Warning: xxxxxxx ....
Jan 3, 2002 6:54:18 PM com.sapmarkets.usermanagement.ClassB.method2 [main] Fatal: Unable to ....

FIG. 7b

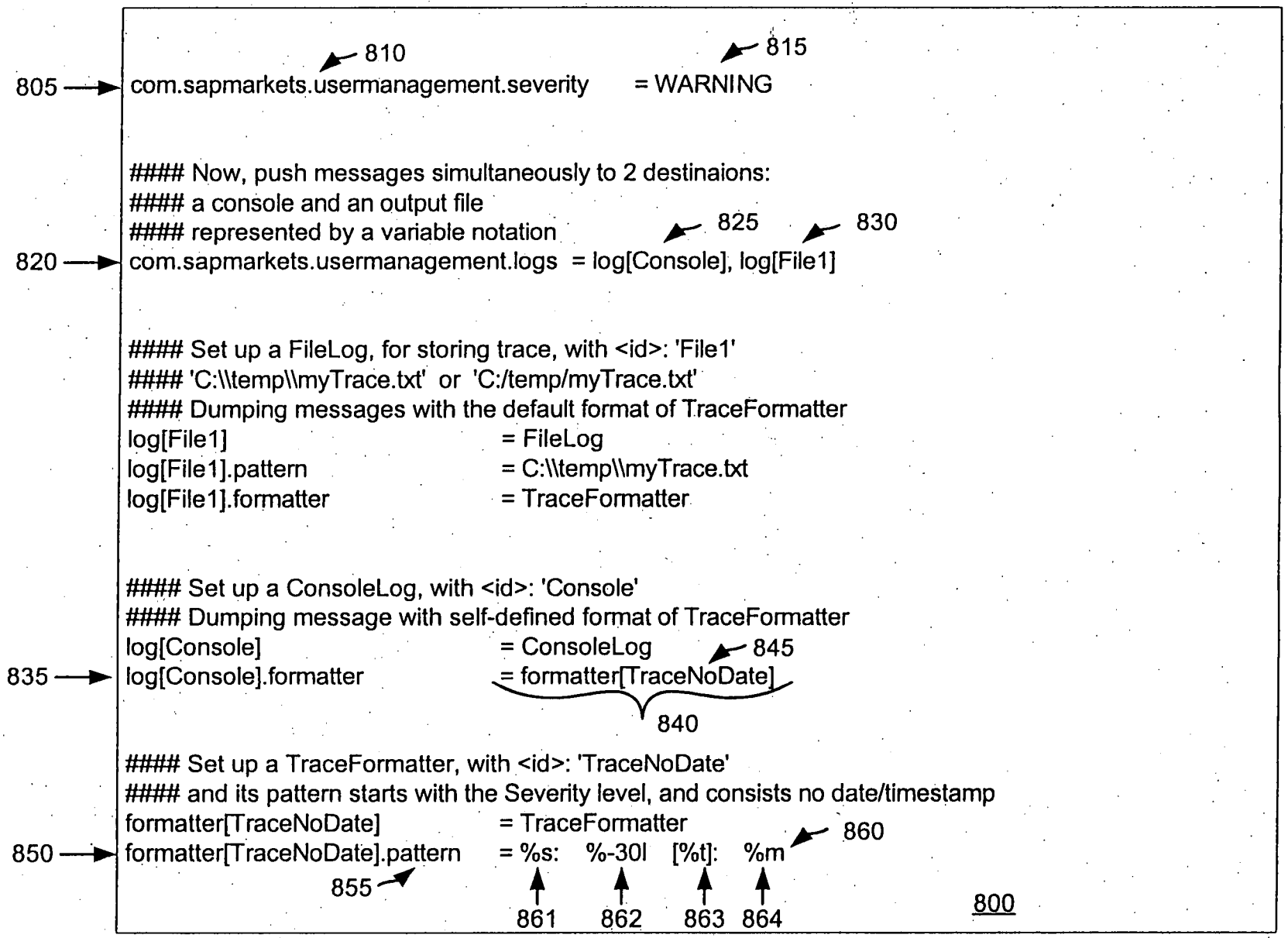
810
815
805
com.sapmarkets.usermanagement.severity = WARNING
Now, push messages simultaneously to 2 destinaions:
a console and an output file
represented by a variable notation
825
830
820
com.sapmarkets.usermanagement.logs = log[Console], log[File1]
Set up a FileLog, for storing trace, with <id>: 'File1'
'C:\\temp\\myTrace.txt' or 'C:/temp/myTrace.txt'
Dumping messages with the default format of TraceFormatter
log[File1] = FileLog
log[File1].pattern = C:\\temp\\myTrace.txt
log[File1].formatter = TraceFormatter
Set up a ConsoleLog, with <id>: 'Console'
Dumping message with self-defined format of TraceFormatter
log[Console] = ConsoleLog
845
835
log[Console].formatter = formatter[TraceNoDate]
840
Set up a TraceFormatter, with <id>: 'TraceNoDate'
and its pattern starts with the Severity level, and consists no date/timestamp
formatter[TraceNoDate] = TraceFormatter
860
850
formatter[TraceNoDate].pattern = %s: %-30l [%t]: %m
855
861
862
863
864
800

FIG. 8

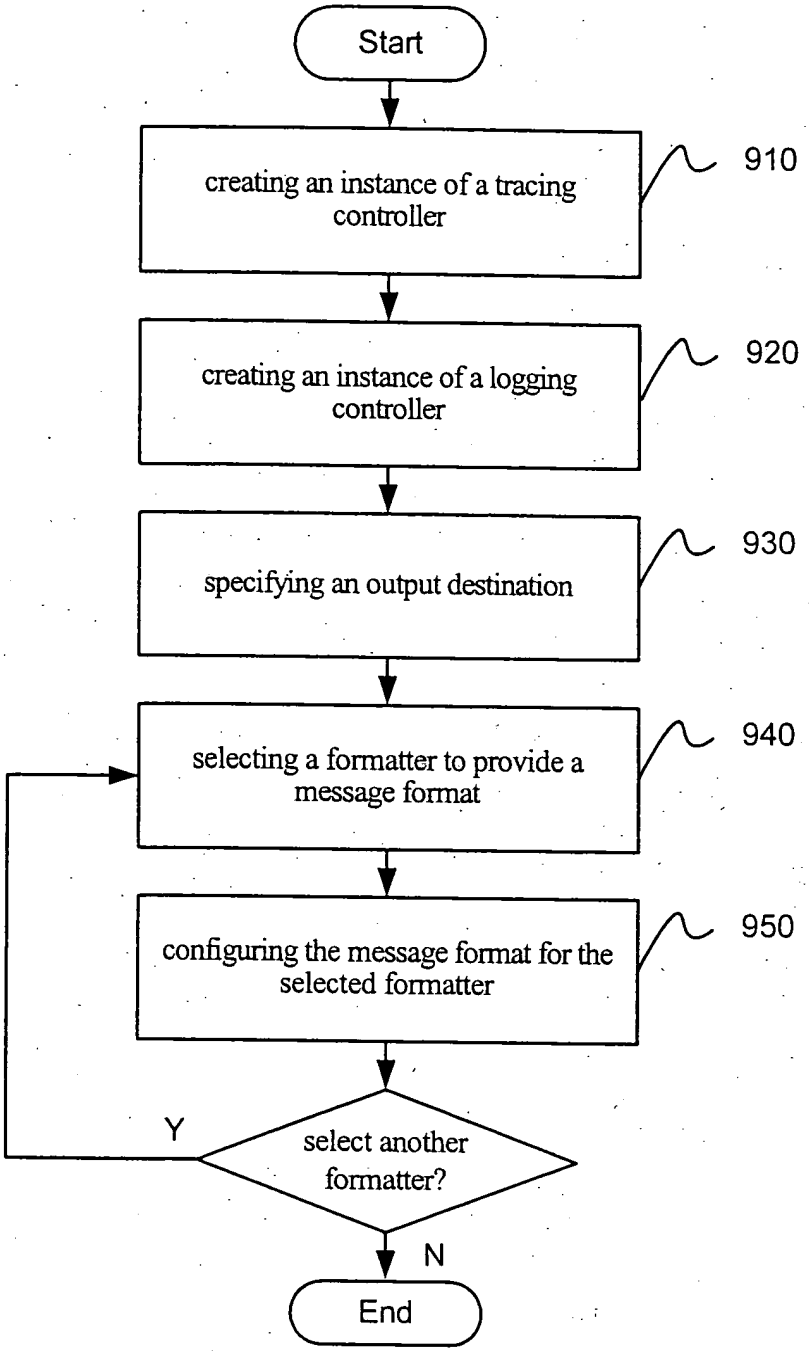
Start
creating an instance of a tracing controller
910
creating an instance of a logging controller
920
specifying an output destination
930
selecting a formatter to provide a message format
940
configuring the message format for the selected formatter
950
select another formatter?
Y
N
End

FIG. 9

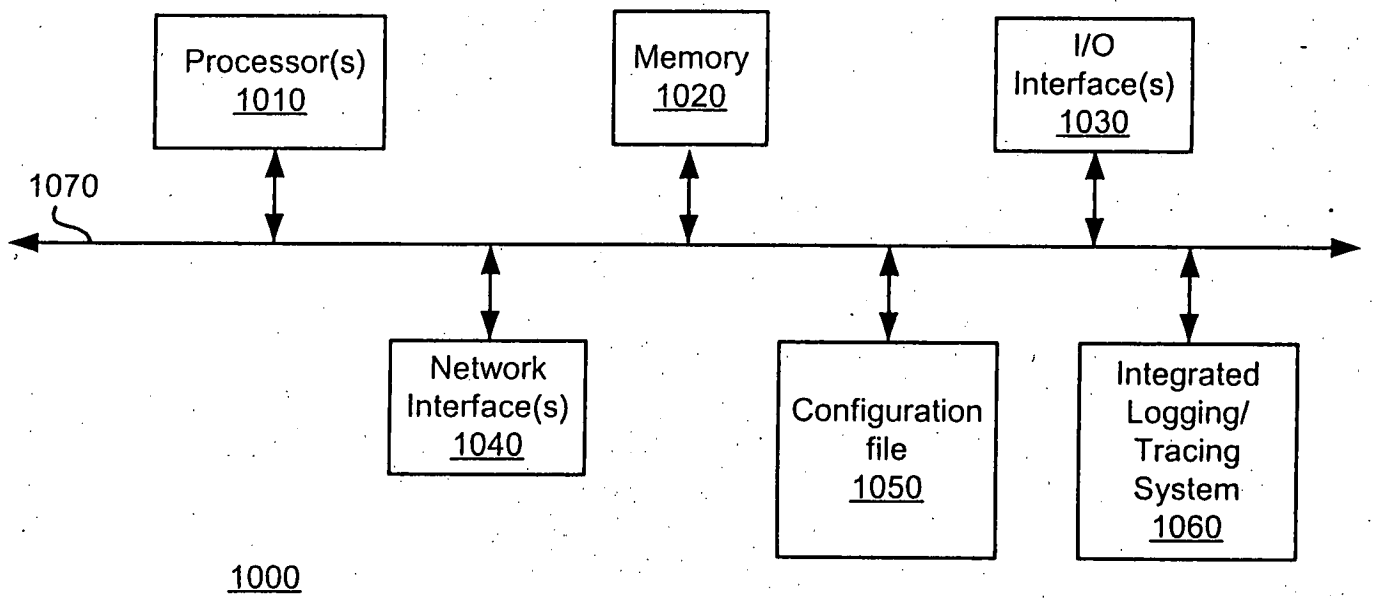
Processor(s)
1010
Memory
1020
I/O
Interface(s)
1030
1070
Network
Interface(s)
1040
Configuration
file
1050
Integrated
Logging/
Tracing
System
1060
1000

FIG. 10